VIRTUAL REVIEW OF LARGE SCALE IMAGE ON 3D DISPLAY

SARTON J. [1] - COURILLEAU N. [1,2] - HÉRARD A-S. [3] - DELZESCAUX T. [3] - REMION Y. [1] - LUCAS L. [1]

1. Reims University (URCA), CreSTIC, France - jonathan.sarton@univ-reims.fr 2, Neoxia, Paris, France - nicolas.courilleau@neoxia.com
3, CEA-CNRS-UMR 9199, MIRCen, France.

Motivations

Nowadays, **large scale images** allow pathologists to perform reviews, using computer workstation (**Virtual Microscopy**) instead of microscope. However, these technologies introduce some issues:

- The very large amount of data generated,
- The loss of the 3D perspective given by real microscope.

To compensate these issues we propose a system to visualize images using 3D **autostereoscopic displays** and **GPU out-of-core data management** in order to provide an interactive navigation and a better user experience.

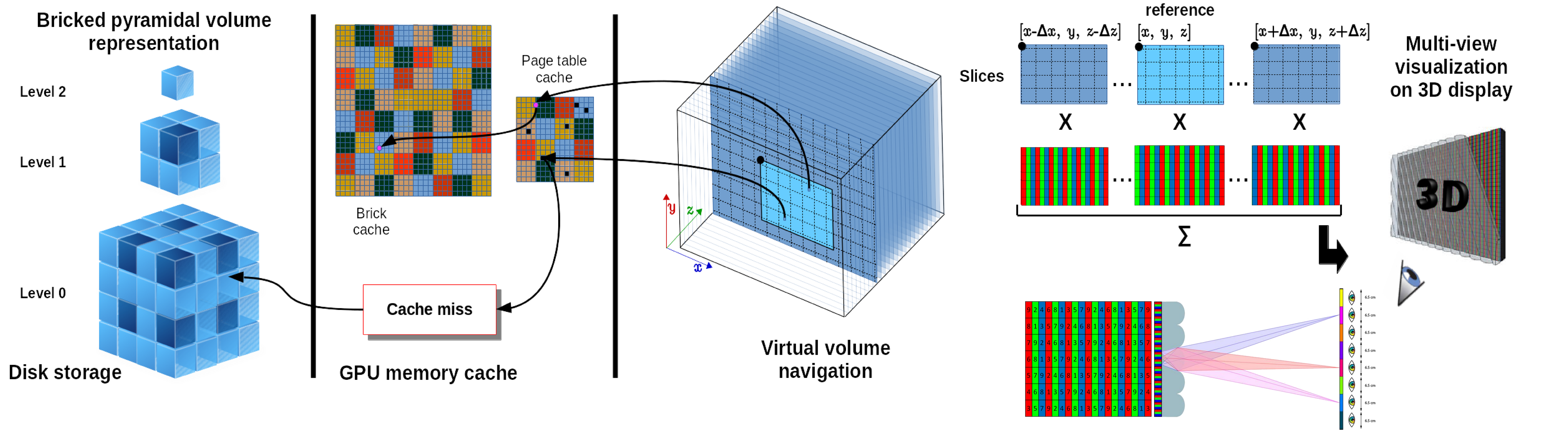
DATA REPRESENTATION

The large 3D volume is preprocessed to create a multi-scale representation (**3D mipmap**) divided and stored as small **bricks** (small compressed 3D volume e.g. 32^3 voxels) on a large data storage.
The bricks are streamed on-demand to the visualization stage on the GPU. The level of resolution is adapted to the current screen resolution or to the desired level of detail.

VIRTUAL VOLUME NAVIGATION

The navigation is performed, from the gpu, in a **virtual volumes** that represent the whole stack images from the lowest to the highest resolution. In order to get a voxel value 2 parameters are required:

- l which is the level of resolution we are looking at.
- $p = [x, y, z]$ where $x, y, z \in [0, 1[$ the 3D **normalize position** in the virtual volume of the resolution l.



OUT-OF-CORE DATA MANAGEMENT

An out-of-core data management is used to address the whole large volume from the GPU. It is based on a multi-level, multi-resolution **page table** hierarchy with a **cache mechanism** entirely managed on the GPU. One entry of page table virtualize a large block of data through one or more level of virtualization(depend on the size of the volume).

Using this hierarchy it is possible to address several peta bytes of data.

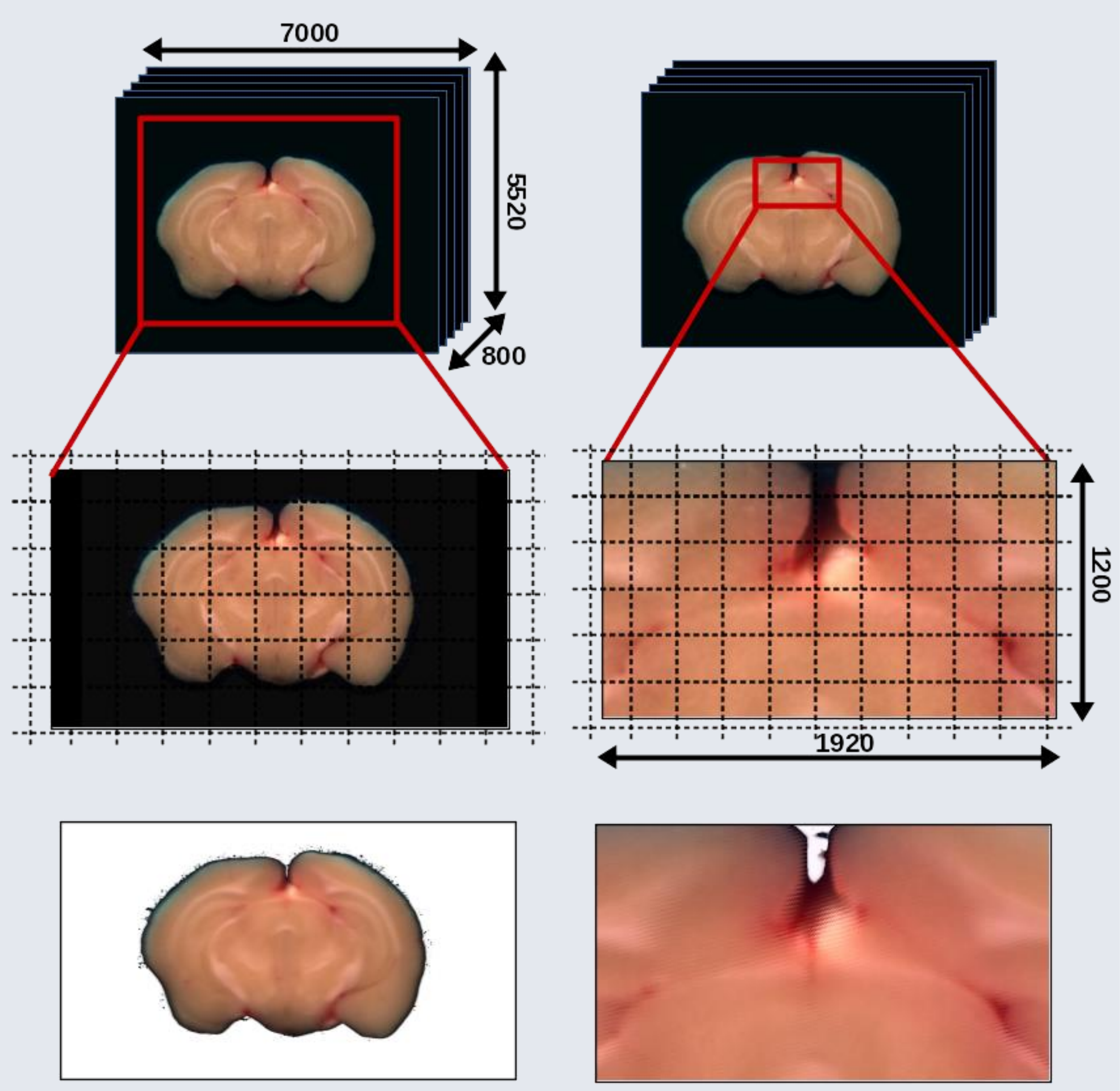
AUTOSTEREOSCOPIC VIEW CREATION

An autostereoscopic frame is composed of N views (N = number of point of view of the screen). To create a frame, a **reference view** is selected and the others are created from this one by applying a:

- Δ_x for the horizontal disparity.
- Δ_z for the depth perception.

A multiplexing filter is applied to each view before summing it up. A dynamic transfer function combined to alpha-blending is used during pixel compositing to determine a desired transparency on a specific range of intensity or color.

RESULTS



Mouse brain block-face volume **(~86 GB)** display at **~30 FPS** on Intel i7 6700HQ @ 2.6GHz, Geforce GTX960M and an SSD in PCIe.

GPU memory usage:

$$\left(\frac{H}{b}+1\right) * \left(\frac{W}{b}+1\right) * 2 * (b^3 * p)$$

With: H & W = screen height and width.
p = p-bytes encoded pixels volume.
b^3 = pixels bricks size.

Few hundred MB = Largely ok for modern GPUs.
Output sensitive algorithm: do not depend of the input size of the volume.

Time analysis:

- One multi-view frame building: ~30ms depends on the size and the number of view of the 3D screen.
- Brick loading: Depends on the brick size. This does not affect FPS since brick loading are asynchronous.

This work is supported by the French national funds (PIA2'program) under contract No. P112331-3422142 (3DNS project). We would like to thank the three clusters (Cap Digital and Systematic for ICT and Medicen for Health) that support this project.